```
DDDDDDDDDDD      UUU          UUU  MMM          MMM  PPPPPPPPPPP
DDDDDDDDDDD      UUU          UUU  MMM          MMM  PPPPPPPPPPP
DDDDDDDDDDD      UUU          UUU  MMM          MMM  PPPPPPPPPPP
DDD       DDD    UUU          UUU  MMMMMM    MMMMMM  PPP        PPP
DDD       DDD    UUU          UUU  MMMMMM    MMMMMM  PPP        PPP
DDD       DDD    UUU          UUU  MMMMMM    MMMMMM  PPP        PPP
DDD       DDD    UUU          UUU  MMM  MMM  MMM     PPP        PPP
DDD       DDD    UUU          UUU  MMM   MMM   MMM   PPP        PPP
DDD       DDD    UUU          UUU  MMM    MMM  MMM   PPP        PPP
DDD       DDD    UUU          UUU  MMM          MMM  PPPPPPPPPPP
DDD       DDD    UUU          UUU  MMM          MMM  PPPPPPPPPPP
DDD       DDD    UUU          UUU  MMM          MMM  PPP
DDD       DDD    UUU          UUU  MMM          MMM  PPP
DDD       DDD    UUU          UUU  MMM          MMM  PPP
DDD       DDD    UUU          UUU  MMM          MMM  PPP
DDD       DDD    UUU          UUU  MMM          MMM  PPP
DDDDDDDDDDD      UUUUUUUUUUUUUUUU  MMM          MMM  PPP
DDDDDDDDDDD      UUUUUUUUUUUUUUUU  MMM          MMM  PPP
DDDDDDDDDDD      UUUUUUUUUUUUUUUU  MMM          MMM  PPP
```

```
DDDDDDD    UU      UU  MM      MM  PPPPPPP   FFFFFFFFFF  AAAAAA      000000   LL          NN        NN
DDDDDDD    UU      UU  MM      MM  PPPPPPP   FFFFFFFFFF  AAAAAA      000000   LL          NN        NN
DD     DD  UU      UU  MMMM  MMMM  PP     PP  FF          AA    AA  00    00  LL          NN        NN
DD     DD  UU      UU  MMMM  MMMM  PP     PP  FF          AA    AA  00    00  LL          NN        NN
DD     DD  UU      UU  MM  MM  MM  PP     PP  FF          AA    AA  00    00  LL          NNNN      NN
DD     DD  UU      UU  MM  MM  MM  PP     PP  FF          AA    AA  00    00  LL          NNNN      NN
DD     DD  UU      UU  MM      MM  PPPPPPP   FFFFFFF      AA    AA  00    00  LL          NN  NN    NN
DD     DD  UU      UU  MM      MM  PPPPPPP   FFFFFFF      AAAAAAAAAA  00    00  LL        NN  NN    NN
DD     DD  UU      UU  MM      MM  PP        FF          AAAAAAAAAA  00    00  LL          NN    NNNN
DD     DD  UU      UU  MM      MM  PP        FF          AA    AA  00    00  LL            NN    NNNN
DD     DD  UU      UU  MM      MM  PP        FF          AA    AA  00    00  LL            NN        NN
DD     DD  UU      UU  MM      MM  PP        FF          AA    AA  00        LL            NN        NN
DDDDDDD    UUUUUUUUUU  MM      MM  PP        FF          AA    AA    000000   LLLLLLLLLL   NN        NN
DDDDDDD    UUUUUUUUUU  MM      MM  PP        FF          AA    AA    000000   LLLLLLLLLL   NN        NN
```

```
LL          IIIIII    SSSSSSSS
LL          IIIIII    SSSSSSSS
LL            II          SS
LL            II          SS
LL            II          SS
LL            II          SS
LL            II      SSSSSS
LL            II      SSSSSS
LL            II          SS
LL            II          SS
LL            II          SS
LL            II          SS
LLLLLLLLLL  IIIIII    SSSSSSSS
LLLLLLLLLL  IIIIII    SSSSSSSS
```

```
0000          1                .TITLE  DUMP$FAO_LINE, format one line
0000          2                .IDENT  'V04-000'
0000          3        ;---
0000          4
0000          5        ;******************************************************************
0000          6        ;*                                                                *
0000          7        ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                       *
0000          8        ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.        *
0000          9        ;*  ALL RIGHTS RESERVED.                                          *
0000         10        ;*                                                                *
0000         11        ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000         12        ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000         13        ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0000         14        ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000         15        ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0000         16        ;*  TRANSFERRED.                                                   *
0000         17        ;*                                                                *
0000         18        ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000         19        ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000         20        ;*  CORPORATION.                                                   *
0000         21        ;*                                                                *
0000         22        ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000         23        ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.        *
0000         24        ;*                                                                *
0000         25        ;*                                                                *
0000         26        ;******************************************************************
0000         27
0000         28        ;++
0000         29        ; FACILITY:
0000         30        ;       File dump utility.
0000         31        ;
0000         32        ; ABSTRACT:
0000         33        ;       This module contains the routine to format one line.
0000         34        ;
0000         35        ; ENVIRONMENT:
0000         36        ;       VAX native, user mode.
0000         37        ;
0000         38        ; AUTHOR:  Benn Schreiber, Stephen Zalewski      CREATION DATE:  22-Jun-1981
0000         39        ;
0000         40        ; MODIFIED BY:
0000         41        ;
0000         42        ;       V02-001 MLJ0033         Martin L. Jack, 23-Aug-1981  9:48
0000         43        ;               Minor cleanup to finish implementation.
0000         44        ;
0000         45        ;**
0000         46
0000         47
00000000     48                .PSECT  $CODE$,EXE,NOWRT
```

```
                              0000  50              .SBTTL  DUMP$FAO_LINE, format one line
                              0000  51      ;
                              0000  52      ; Format one line of dump listing
                              0000  53      ;
                              0000  54      ; Inputs:
                              0000  55      ;
                              0000  56      ;       04(ap) = pointer to data to be dumped
                              0000  57      ;       08(ap) = number of entries per line
                              0000  58      ;       12(ap) = size of one entry
                              0000  59      ;       16(ap) = byte-offset value for side of line
                              0000  60      ;       20(ap) = number of entries in buffer
                              0000  61      ;       24(ap) = 0: longword, 1: word, 2: byte
                              0000  62      ;       28(ap) = address of descriptor for FAO control string
                              0000  63      ;       32(ap) = address of descriptor for output buffer
                              0000  64      ;
                    003C      0000  65              .entry  dump$fao_line,^M<R2,R3,R4,R5>
      53    08 AC    7D       0002  66              movq    8(ap),r3                ; R3 = entries per line, R4 = size of entry
            54 53    C4       0006  67              mull2   r3,r4                   ; compute number of bytes this line
            10 AC    DD       0009  68              pushl   16(ap)                  ; push index to print on right hand side
            04 AC    DD       000C  69              pushl   4(ap)                   ; push buffer address
            54       DD       000F  70              pushl   r4                      ; push number of bytes
      52    14 AC    D0       0011  71              movl    20(ap),r2               ; get number of entries in line
   08 AC    52       D1       0015  72              cmpl    r2,8(ap)                ; see if more than one line's worth
            04       15       0019  73              bleq    10$                     ; if leq no
      52    08 AC    D0       001B  74              movl    8(ap),r2                ; yes, use max for one line
      51    04 AC    D0       001F  75  10$:        movl    4(ap),r1                ; copy input data pointer
      50    18 AC    D0       0023  76              movl    24(ap),r0               ; get/test field width
            13       13       0027  77              beql    40$                     ; if eql then longwords
            08 50    E8       0029  78              blbs    r0,30$                  ; branch if words
                              002C  79      ;
                              002C  80      ; push bytes onto stack
                              002C  81      ;
      7E    81       9A       002C  82  20$:        movzbl  (r1)+,-(sp)             ; push one byte
         FA 52       F5       002F  83              sobgtr  r2,20$                  ; do them all
            0D       11       0032  84              brb     50$                     ; go call fao
                              0034  85      ;
                              0034  86      ; push words onto stack
                              0034  87      ;
      7E    81       3C       0034  88  30$:        movzwl  (r1)+,-(sp)             ; push one word
         FA 52       F5       0037  89              sobgtr  r2,30$                  ; do them all
            05       11       003A  90              brb     50$                     ; go call fao
                              003C  91      ;
                              003C  92      ; push longwords onto stack
                              003C  93      ;
            81       DD       003C  94  40$:        pushl   (r1)+                   ; push one longword
         FB 52       F5       003E  95              sobgtr  r2,40$                  ; do them all
                              0041  96      ;
                              0041  97      ; call $FAO
                              0041  98      ;
            6E       9F       0041  99  50$:        pushab  (sp)                    ; push address of arg list
         20 AC       DD       0043  100             pushl   32(ap)                  ; push output buffer descr. addr
            6E       DD       0046  101             pushl   (sp)                    ; also for output width
         1C AC       DD       0048  102             pushl   28(ap)                  ; push fao control string addr
00000000'GF 04       FB       004B  103             calls   #4,g^sys$faol           ; call sys$faol to format string
                     04       0052  104             ret
                              0053  105
                              0053  106             .end
```

```
DUMP$FAO_LINE     00000000 RG    01
SYS$FAOL          ********  X     01
```

+-------------------+
! Psect synopsis !
+-------------------+

PSECT name	Allocation		PSECT No.		Attributes										
. ABS .	00000000	(0.)	00 (	0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
$CODE$	00000053	(83.)	01 (	1.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE

+------------------------------+
! Performance indicators !
+------------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:01.09
Command processing	133	00:00:00.36	00:00:04.77
Pass 1	72	00:00:00.28	00:00:02.47
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	36	00:00:00.17	00:00:01.07
Symbol table output	2	00:00:00.00	00:00:00.00
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	275	00:00:00.91	00:00:09.42

The working set limit was 900 pages.
1493 bytes (3 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 2 non-local and 5 local symbols.
106 source lines were read in Pass 1, producing 14 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

+--------------------------------+
! Macro library statistics !
+--------------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:DUMPFAOLN/OBJ=OBJ$:DUMPFAOLN MSRC$:DUMPFAOLN/UPDATE=(ENH$:DUMPFAOLN)

DTSMAIN
LIS

DUMP
LIS

DUMPFILE
LIS

DTSMSG
LIS

DTSPARSE
LIS

DTRTEST
LIS

DUMPFAOLN
LIS

DUMPRE
REQ

DUMP

DUMP
MAP

DUMPHEADE
LIS

DTSAST
LIS

DTSTEST
LIS